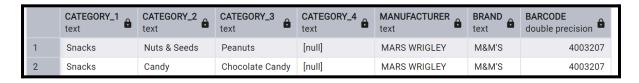
# Fetch Take home assessment Part 2

For this exercise, I connected the jupyter notebook to the PostgresQL database server to load the 3 datasets into the PostgresQL environment.

# Checks on join of transactions and products -

- The number of rows increased after performing a left join on transactions with the products dataset which indicates there must've been a duplication of records.
- On checking, I came across two RECEIPT IDs that duplicated after join operation
  - RECEIPT\_ID "2f984bec-0243-475c-9e1e-ed9933f70fd1" and "b59e828a-2040-4188-aa9f-70b53cdc854d" associated with BARCODE "4003207"
  - BARCODE "4003207" is associated with two values in CATEGORY 2 and CATEGORY 3.



#### Assumptions made -

- Due to the lack of any information/context on the above BARCODE, I have chosen the second record as the true record for "4003207" BARCODE mentioned above and removed the first record from the dataset to avoid any duplication of records. The second record seems more appropriate for M&M'S as it is a candy.
- FINAL\_SALE value of ' ' (blank) replaced with 0 for aggregations
- FINAL QUANTITY value of 'zero' replaced with 0 for aggregations

```
clean_trans_df.loc[clean_trans_df['FINAL_SALE'] == ' ', 'FINAL_SALE']= '0.00'
clean_trans_df.loc[clean_trans_df['FINAL_QUANTITY'] == 'zero', 'FINAL_QUANTITY']= '0.00'
```

#### Code for checks on duplication of records after join operation-

# #Count of records per receipt id

WITH receipt\_count\_table AS (
SELECT "RECEIPT\_ID", COUNT(\*) AS count1
FROM transactions
GROUP BY "RECEIPT\_ID"),

# #Count of records per receipt id after join of transactions and products

join\_receipt\_count\_table AS (
SELECT t."RECEIPT\_ID", COUNT(\*) AS count2
FROM
transactions t
LEFT JOIN products p
ON t."BARCODE" = p."BARCODE"
GROUP BY t."RECEIPT\_ID"),

# #Joining the the two above CTEs and performing a check on counts at RECEIPT\_ID level

receipt\_count\_check\_table AS (
SELECT \*, (count1 - count2) AS countcheck
FROM
receipt\_count\_table LEFT JOIN join\_receipt\_count\_table
ON receipt\_count\_table."RECEIPT\_ID" = join\_receipt\_count\_table."RECEIPT\_ID")
#Extracting ids of receipt that duplicated and see the associated BARCODE

SELECT\* FROM receipt\_count\_check\_table WHERE countcheck <>0;

#Query to remove the row not required for the '4003207' BARCODE

**DELETE FROM products** 

WHERE "BARCODE" = '4003207' AND "CATEGORY 2" = 'Nuts & Seeds';

#### Close ended -

1. What are the top 5 brands by receipts scanned among users 21 and over? Filtered out "null" brand value from this question. The top 5 brands are Dove, Nerds Candy, Hershey's, Coca-Cola and Great Value

Query - The below code uses extract function and case statement to take users 21 and over based on the current date (Dynamic in nature). Please note that there are more brands with 2 receipts scanned other than the ones shown below in the screenshot. Currently the result shows top 5 just based on how the SQL environment prints them sorted on number of receipts scanned but there should be more criterias that can be added to determine true "top" 5 amongst the brands that have 2 receipts scanned.

SELECT p."BRAND", COUNT(DISTINCT "RECEIPT\_ID") AS receipt\_count #selecting Brand and distinct count of receipt ID

FROM transactions t

LEFT JOIN users u ON t."USER\_ID" = u."ID" #performing a left join on users dataset

LEFT JOIN products p ON t."BARCODE" = p."BARCODE" #performing a left join on products dataset

WHERE p."BRAND" IS NOT NULL #filtering out null brand values

AND (EXTRACT(YEAR FROM CURRENT\_DATE) - EXTRACT(YEAR FROM "BIRTH\_DATE"))

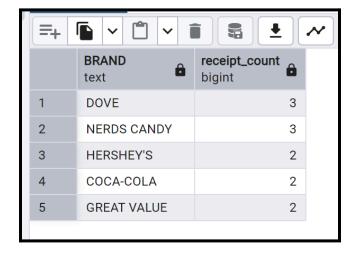
- CASE

WHEN EXTRACT(MONTH FROM CURRENT\_DATE) < EXTRACT(MONTH FROM "BIRTH\_DATE")
OR (EXTRACT(MONTH FROM CURRENT\_DATE) = EXTRACT(MONTH FROM "BIRTH\_DATE")
AND EXTRACT(DAY FROM CURRENT\_DATE) < EXTRACT(DAY FROM "BIRTH\_DATE"))
THEN 1 ELSE 0

END >= 21 #calculating exact age based on current date and filtering for 21 and over GROUP BY p."BRAND"

ORDER BY receipt\_count DESC

LIMIT 5;



2. What is the percentage of sales in the Health & Wellness category by generation? Defined generation based on birth\_date. 24% of Total Sales are associated with records that do not have a birth date specified and 99% of sales within Health & Wellness are associated with records that do not have a birth date specified.

```
Query -
SELECT
CASE #creating generation column based on age range
       WHEN u.age IS NULL THEN 'Birth date missing'
       WHEN u.age<=12 THEN 'Gen Alpha'
       WHEN u.age>12 AND u.age<=28 THEN 'Gen Z'
       WHEN u.age>28 AND u.age<=44 THEN 'Millennials'
       WHEN u.age>44 AND u.age<=60 THEN 'Gen X'
       WHEN u.age>60 THEN 'Baby Boomer and earlier'
END AS generation,
ROUND(SUM(t."FINAL SALE")::NUMERIC.0) AS total sales, #total sales in transactions dataset
ROUND(SUM(CASE WHEN p."CATEGORY 1" = 'Health & Wellness' THEN t."FINAL SALE" ELSE 0
END)::NUMERIC,0) AS health wellness sales, #total health & wellness category sales
ROUND(
(SUM(CASE
               WHEN p."CATEGORY 1" = 'Health & Wellness' THEN t."FINAL SALE"
               ELSE 0
       END)*100)::NUMERIC
       /(SELECT SUM(t."FINAL SALE")::NUMERIC
       FROM transactions t
       LEFT JOIN products p ON t."BARCODE" = p."BARCODE")
,2)AS percentage of total, #health & wellness sales as a % of total sales
ROUND(
(SUM(CASE
               WHEN p."CATEGORY 1" = 'Health & Wellness' THEN t."FINAL SALE"
               ELSE 0
       END)*100)::NUMERIC
       /(SELECT SUM(t."FINAL SALE")::NUMERIC
       FROM transactions t
       LEFT JOIN products p ON t."BARCODE" = p."BARCODE"
       WHERE p."CATEGORY 1"='Health & Wellness')
,2) AS percentage of health wellness #%of total health & wellness sales
FROM
transactions t
LEFT JOIN (SELECT *, (EXTRACT(YEAR FROM CURRENT DATE) - EXTRACT(YEAR FROM "BIRTH DATE"))

    CASE

    WHEN EXTRACT(MONTH FROM CURRENT_DATE) < EXTRACT(MONTH FROM "BIRTH_DATE")
    OR (EXTRACT(MONTH FROM CURRENT DATE) = EXTRACT(MONTH FROM "BIRTH DATE")
    AND EXTRACT(DAY FROM CURRENT DATE) < EXTRACT(DAY FROM "BIRTH DATE"))
    THEN 1 ELSE 0
    END AS age
FROM users) u #using subquery to add an Age column in users database for generation grouping
ON t."USER ID" = u."ID" #performing a left join on users dataset
LEFT JOIN products p ON t."BARCODE" = p."BARCODE" #performing a left join on products dataset
GROUP BY generation ORDER BY total sales DESC; #showing the metrics by generation
```

#### Result -

=+	Showing rows: 1 to 5						
	generation text	total_sales numeric	health_wellness_sales numeric	percentage_of_total numeric	percentage_of_health_wellness numeric		
1	Birth date missing	170298	41145	24.04	99.54		
2	Millennials	323	59	0.03	0.14		
3	Baby Boomer and earlier	276	84	0.05	0.20		
4	Gen X	264	46	0.03	0.11		
5	Gen Z	22	0	0.00	0.00		

# Open ended -

3. Which is the leading brand in the Dips & Salsa Category? I have categorized the leading brand in the Dips & Salsa Category based on the number of receipts scanned, number of stores the brand's product is bought in, total sales & quantity. The brand 'TOSTITOS' seems to be the leading brand across all metrics.

#### Query -

SELECT p. "BRAND",

COUNT(DISTINCT t."RECEIPT\_ID") AS receipt\_count, #distinct count of receipts scanned

ROUND(SUM(t."FINAL\_SALE")::NUMERIC,0) AS total\_sales, #total sales

COUNT(DISTINCT t."STORE\_NAME") AS total\_stores, #number of stores

SUM(t."FINAL\_QUANTITY") AS total\_quantity #total quantity

FROM transactions t

LEFT JOIN products p ON t."BARCODE" = p."BARCODE"

WHERE p. "CATEGORY\_2" = 'Dips & Salsa' AND p. "BRAND" IS NOT NULL #filtering for Dips & Salsa category and filtering out null brand

GROUP BY p."BRAND"

ORDER BY receipt\_count DESC; #currently ordered by receipt\_count but when ordered by other metrics, TOSTITOS still stays the leader

#### Result-

	BRAND text	receipt_count bigint	total_sales numeric	total_stores bigint	total_quantity double precision
1	TOSTITOS	36	261	16	60
2	PACE	24	119	8	38
3	FRITOS	19	92	9	33
4	DEAN'S DAIRY DIP	17	68	4	22
5	MARKETSIDE	16	103	1	23